

## **DETAILED ACTION**

1. This communication is in response to the Pre-Brief Conference Request filed 22 August 2009.
2. As a result of the Pre-conference Brief filed 22 August 2009 and the Examiner's Amendment stated below, claims 3-5, 7-9, 11-13, 20, 31, 32, 34-36, 38, 39 and 41 (renumbered as 1-18) are allowed.

### ***35 USC § 101 - Clarifications***

3. The computer-readable medium disclosed in claim 39 is interpreted as being analogous to the computer-usuable storage medium disclosed in page 25, lines 1-20 of the Specification filed 24 July 2003. According to the definition of the medium, the examiner construes the medium as being limited to statutory embodiments which meet the requirements of 35 USC 101.

### ***Examiner Amendment***

4. Authorization for this examiner's amendment, listed below, was given in a telephone interview with Marcia Doubet (Reg. No. 40,999) on 12 March 2010.

#### **In the Abstract:**

Please replace the Abstract with the following marked-up replacement Abstract:

-- Techniques are disclosed for enabling a validating parser to interpret a

schema, where that schema may have been extended to multiple levels, and to perform the validation of a structured document according to the extended schema while still allowing a consumer application to request objects or events that are cast at some different level of the schema (including the base schema). Validating the source document according to an extended (i.e., more specific) schema enables improved detection of syntax errors. At the same time, casting objects and events from the parsed document to a level requested by the consumer application allows the consumer to receive only those objects or events for which it is adapted, without requiring the consumer application to include extra code to deal with objects or events it does not recognize in the parser=s output. Preferably, the parser simply discards those objects or events which this consumer is not interested in receiving. --

**In the Claims:**

Please amend claims 4, 7, 8, 13, 31, 32, 33, 36, 37, 39 and 40 as follows:

Claim 4 (currently amended): The method according to Claim 3, wherein the structured document is encoded in Extensible Markup Language ([["]]XML[["]]).

Claim 7 (currently amended): The method according to Claim [[33]] 32, wherein the second schema definition is requested by specifying a schema name of the second schema definition, to which the generated output must adhere.

Claim 8 (currently amended): The method according to Claim [[33]] 32, wherein the second schema definition is requested by specifying a schema name of the second schema definition, indicating that the second schema definition is to be used by the validating parser when generating the output.

Claim 13 (currently amended): A computer-implemented method of casting objects, comprising:

providing a validating parser that is-adapted configured for validating whether syntax elements of an input document conform to a first schema definition identified in the input document while generating output objects, from the validated syntax elements of the input document, that conform to a second schema definition dynamically selected by a consuming application of the generated output objects;

using the validating parser for validating whether the syntax elements of the input document conform to the first schema definition, wherein:

the first schema definition is an extended schema;

using the validating parser, responsive to the validating of the syntax elements, for generating the output objects to conform to the second schema definition, wherein:

the second schema definition is a base schema from which the extended schema was extended, such that the extended schema defines at least one syntax element that is not defined in the base schema; and

the generating further comprises not generating any output object for any of the at least one syntax element that is defined in the extended schema but not

defined in the base schema in order that the generated output objects will conform to the second schema definition; and

providing the generated output objects, by the validating parser, for use by the consuming application.

Claim 31 (currently amended): A computer-implemented method of providing validation and parsing, comprising:

providing a validating parser ~~adapted~~ configured for validating an input document according to a first schema definition identified in the input document while generating output, from the validated input document, according to a second schema definition dynamically selected by a consuming application of the generated output;

validating syntax elements of the input document with the provided validating parser according to the first schema definition, wherein the first schema definition is an extended schema which specifies a syntax definition to which the syntax elements of the input document are to adhere; and

responsive to the validating of the syntax elements, parsing the validated syntax elements to generate the output for the consuming application according to the second schema definition, wherein the second schema definition is a base schema from which the extended schema was extended, thereby suppressing at least one of the validated syntax elements when generating the output for the consuming application, wherein each of the suppressed syntax elements is valid according to the extended schema but is not valid according to the base schema.

Claim 32 (currently amended): A computer-implemented method of applying abstraction by a validating parser, comprising:

using, by a validating parser, a first schema definition for validating syntax elements when parsing syntax of an input document, wherein the first schema definition is identified in the input document; and

omitting, by the validating parser when generating output from the parsed syntax of the input document, each of at least one of the validated syntax elements which is valid according to the first schema definition but is not valid according to a second schema definition for which the output is generated, wherein:

the first schema definition is an extended schema; [[and]]

the second schema definition is a base schema from which the extended schema is extended, such that the extended schema defines at least one syntax element that is not defined in the base schema; and

the second schema definition is dynamically requested, to the validating parser, by an application program for which the output is being generated.

Claim 33 (canceled)

Claim 36 (currently amended): A system for applying abstraction with a validating parser, comprising:

a computer comprising a processor;

a validating parser ~~usable by a~~ executing on the computer; and  
instructions which are ~~executable~~ executed, using [[a]] the processor of the  
computer, to perform:

using, by the validating parser, a first schema definition ~~for validating to~~  
validate syntax elements when parsing syntax of an input document, wherein the first  
schema definition is identified in the input document; and

omitting, by the validating parser when generating output from the parsed  
syntax of the input document, each of at least one of the validated syntax elements  
which is valid according to the first schema definition but is not valid according to a  
second schema definition for which the output is generated, wherein:

the first schema definition is an extended schema; [[and]]

the second schema definition is a base schema from which the  
extended schema is extended, such that the extended schema defines at least one  
syntax element that is not defined in the base schema; and

the second schema definition is dynamically requested, to the  
validating parser, by an application program for which the output is being generated.

#### Claim 37 (canceled)

Claim 39 (currently amended): A computer program product for applying abstraction  
with a validating parser, the computer program product embodied on at least one  
computer-readable medium and comprising computer-readable program code for:

using, by a validating parser, a first schema definition for validating syntax elements when parsing syntax of an input document, wherein the first schema definition is identified in the input document; and

omitting, by the validating parser when generating output from the parsed syntax of the input document, each of at least one of the validated syntax elements which is valid according to the first schema definition but is not valid according to a second schema definition for which the output is generated, wherein:

the first schema definition is an extended schema; [[and]]

the second schema definition is a base schema from which the extended schema is extended, such that the extended schema defines at least one syntax element that is not defined in the base schema; and

the second schema definition is dynamically requested, to the validating parser, by an application program for which the output is being generated.

Claim 40 (canceled)

#### ***Reasons for Allowance***

5. The following is an examiner's statement of reasons for allowance:

In the Final Rejection mailed 25 June 2009, claims 3-5, 7-9, 11-13, 20 and 31-41 were rejected under 35 USC 103(a) based primarily on US Patent No 7,065,742 to Bogdan, US PGPub 2005/0149847 to Chandler and US Patent No 7,458,082 to Slaughter et al.

The claimed invention is directed towards enabling a validating parser to interpret an extended schema, and to perform validation of a structured document according to the extended schema while allowing a consumer application to request objects or events that are cast at a base level of the schema.

The prior art of record, Bogdan, Chandler and Slaughter, does not show, teach or suggest limitations of **providing a validating parser that is configured for validating whether syntax elements of an input document conform to a first schema definition identified in the input document while generating output objects, from the validated syntax elements of the input document, that conform to a second schema definition dynamically selected by a consuming application of the generated output objects; wherein the first schema definition is an extended schema; wherein the second schema definition is a base schema from which the extended schema was extended, such that the extended schema defines at least one syntax element that is not defined in the base schema; and the generating further comprises not generating (omitting/suppressing) any output object for any of the at least one syntax element that is defined in the extended schema but not defined in the base schema in order that the generated output objects will conform to the second schema definition** in combination with the other claimed features.

Referring to Applicant's arguments on pages 1-4 of the Pre-Brief Conference Request, the Applicant argues that the cited prior art fails to teach the concept of not generating any output object for any of the at least one syntax element that is defined in

the extended schema but not defined in the base schema in order that the generated output objects will conform to the base schema definition. Applicant's arguments have been fully considered and are persuasive.

An updated search for prior art on the EAST database and on domains (NPL-Google and ACM) has been conducted. The prior art searched and investigated in the database and domains does not fairly teach or suggest the teaching of the claimed subject matter as described above and reflected by the combined elements in independent claims 13, 31, 32, 36 and 39. Dependent claims 3-5, 7-9, 11-13, 20, 31, 32, 34-36, 38, 39 and 41 are indicated as being allowable for the same reasons stated above in regards to the independent claims.

6. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

***Contact Information***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KIMBERLY LOVEL whose telephone number is (571)272-2750. The examiner can normally be reached on 9:00 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/John R. Cottingham/  
Supervisory Patent Examiner, Art Unit 2167

/Kimberly Lovel/  
Examiner  
Art Unit 2167

12 March 2010  
/KL/